

On Refinements of Boolean and Parametric Modal Transition Systems

Jan Křetínský^{1,2} and Salomon Sickert¹

¹ Institut für Informatik, Technische Universität München, Germany

² Faculty of Informatics, Masaryk University, Brno, Czech Republic

Abstract. We consider the extensions of modal transition systems (MTS), namely Boolean MTS and parametric MTS and we investigate the refinement problems over both classes. Firstly, we reduce the problem of modal refinement over both classes to a problem solvable by a QBF solver and provide experimental results showing our technique scales well. Secondly, we extend the algorithm for thorough refinement of MTS providing better complexity than via reductions to previously studied problems. Finally, we investigate the relationship between modal and thorough refinement on the two classes and show how the thorough refinement can be approximated by the modal refinement.

1 Introduction

Due to the ever increasing complexity of software systems and their reuse, component-based design and verification have become crucial. Therefore, having a specification formalism that supports *component-based* development and *step-wise refinement* is very useful. In such a framework, one can start from an initial specification, proceed with a series of small and successive refinements until eventually a specification is reached from which an implementation can be extracted directly. In each refinement step, we can replace a single component of the current specification with a more concrete/implementable one. The correctness of such a step should follow from the correctness of the refinement of the replaced component, so that the methodology supports *compositional* verification.

Modal transition systems (MTS) were introduced by Larsen and Thomsen [LT88] in order to obtain an operational, yet expressive and manageable specification formalism meeting the above properties. Their success resides in natural combination of two features. Firstly, it is the simplicity of labelled transition systems, which have proved appropriate for behavioural description of systems as well as their compositions; MTS as their extension inherit this appropriateness. Secondly, as opposed to e.g. temporal logic specifications, MTS can be easily *gradually refined* into implementations while preserving the desired behavioural properties. In this work, we focus on checking the refinement between MTS and also their recent extensions.

The formalism of MTS has proven to be useful in practice. Industrial applications are as old as [Bru97] where MTS have been used for an air-traffic

system at Heathrow airport. Besides, MTS are advocated as an appropriate base for interface theories in [RBB⁺09] and for product line theories in [Nym08]. Further, MTS based software engineering methodology for design via merging partial descriptions of behaviour has been established in [UC04]. Moreover, the tool support is quite extensive, e.g. [BLS95,DFFU07,BML11,BCK11].

MTS consist of a set of states and two transition relations. The *must* transitions prescribe which behaviour has to be present in every refinement of the system; the *may* transitions describe the behaviour that is allowed, but need not be realized in the refinements. This allows for underspecification of non-critical behaviour in the early stage of design, focusing on the main properties, verifying them and sorting out the details of the yet unimplemented non-critical behaviour later.

Over the years, many extensions of MTS have been proposed. While MTS can only specify whether or not a particular transition is required, some extensions equip MTS with more general abilities to describe what *combinations* of transitions are possible. Disjunctive MTS (DMTS) [LX90] can specify that at least one of a given set of transitions is present. One selecting MTS [FS08] allow to choose exactly one of them. Boolean MTS (BMTS) [BKL⁺11] cover all Boolean combinations of transitions. The same holds for acceptance automata [Rac07] and Boolean formulae with states [BDF⁺], which both express the requirement by listing all possible sets instead of a Boolean formula. Parametric MTS (PMTS) [BKL⁺11] add parameters on top of it, so that we can also express persistent choices of transitions and relate possible choices in different parts of a system. This way, one can model hardware dependencies of transitions and systems with prices [BKL⁺12].

Our contribution In this paper, we investigate extensions of MTS with respect to two notions of refinement. The *modal refinement* is a syntactically defined notion extending on the one hand bisimulation and on the other hand simulation. Similarly to bisimulation having a counterpart in trace equivalence, here the counterpart of modal refinement is the *thorough refinement*. It is the corresponding semantically defined notion relating (by inclusion) the sets of implementations of the specifications.

We focus both on theoretical and practical complexity of the refinement problems. While modal refinement on MTS and disjunctive MTS can be decided in polynomial time, on BMTS and PMTS it is higher in the polynomial hierarchy (Π_2 and Π_4 , respectively). The huge success of SAT and also QBF solvers inspired us to reduce these refinement problems to problems solvable by a QBF solver. We have also performed experimental results showing that this solution scales well in the size of the system as well as in the number of parameters, while a direct naive solution is infeasible.

Further, we extend the decision algorithm for thorough refinement checking over MTS [BKLS12] and DMTS [BCK10] to the setting of BMTS and PMTS. We show how PMTS can be translated to BMTS and BMTS can then be transformed to DMTS. As we can decide the problem on DMTS in EXPTIME, this shows decidability for BMTS and PMTS, but each of the translations is inevitably

exponential. However, we show better upper bounds than doubly and triply exponential. To this end, we give also a direct algorithm for showing the problem is in NEXPTIME for BMTS and 2-EXPTIME for PMTS.

Since the thorough refinement is EXPTIME-hard for already MTS, it is harder than the modal refinement, which is in P for DMTS and in Π_4 for PMTS. Therefore, we also investigate how the thorough refinement can be approximated by the modal refinement. While underapproximation is easy, as modal refinement implies thorough refinement, overapproximation is more difficult. Here we extend our method of the deterministic hull for MTS [BKLS09] to both BMTS and PMTS. We prove that for BMTS modal and thorough refinements coincide if the refined system is deterministic, which then yields an overapproximation via the deterministic hull. Finally, in the case with PMTS, we need to overapproximate the behaviour dependent on the parameters, because the coincidence of the refinements on deterministic systems fails for PMTS.

Our contribution can be summarized as follows:

- We reduce the problem of modal refinement over BMTS and PMTS to a problem solvable by a QBF solver. We provide promising experimental results showing this solution scales well.
- We extend the algorithm for thorough refinement on MTS and DMTS to BMTS and PMTS providing better complexity than via translation of these formalisms to DMTS. This also shows (together with results on modal refinement) that we can make use of the more compact representation used in the formalisms of BMTS and PMTS.
- We investigate the relationship between modal and thorough refinement on BMTS and PMTS. We introduce approximation methods for the thorough refinement on BMTS and PMTS through the modal refinement.

Related work There are various other approaches to deal with component *refinements*. They range from subtyping [LW94] over Java modelling language [JP01] to interface theories close to MTS such as interface automata [dAH01]. Similarly to MTS, interface automata are behavioural interfaces for components. However, their composition works very differently. Furthermore, its notion of refinement is based on alternating simulation [AHKV98], which has been proved strictly less expressive than MTS refinement—actually coinciding on a subclass of MTS—in the paper [LNW07], which combines MTS and interface automata based on I/O automata [Lyn88]. The compositionality of this combination is further investigated in [RBB⁺11].

Further, opposite to the design of correct software where an abstract verified MTS is transformed into a concrete implementation, one can consider checking correctness of software through *abstracting* a concrete implementation into a coarser system. The use of MTS as abstractions has been advocated e.g. in [GHJ01]. While usually overapproximations (or underapproximations) of systems are constructed and thus only purely universal (or existential) properties can be checked, [GHJ01] shows that using MTS one can check mixed formulae (arbitrarily combining universal and existential properties) and, moreover, at the

same cost as checking universal properties using traditional conservative abstractions. This advantage has been investigated also in the context of systems equivalent or closely related to MTS [HJS01,DGG97,Nam03,DN04,CGLT09,GNRT10].

MTS can also be viewed as a fragment of mu-calculus that is “graphically representable” [BL90,BDF⁺]. The graphical representability of a variant of alternating simulation called covariant-contravariant simulation has been recently studied in [AFdFE⁺11].

Outline of the paper In Section 2, we recall the formalism of MTS and the extensions discussed. Further, in Section 3, we recall the modal refinement problem. We reduce it to a QBF problem in Section 4. In Section 5, we give a solution to the thorough refinement problems. Section 6 investigates the relationship of the two refinements and how modal refinement can approximate the thorough refinement. We conclude in Section 7.

2 Modal Transition Systems and Boolean and Parametric Extensions

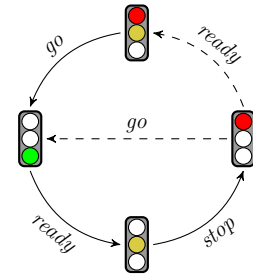
In this section, we introduce the studied formalisms of modal transition systems and their Boolean and parametric extensions. We first recall the standard definition of MTS:

Definition 2.1. A modal transition system (MTS) over an action alphabet Σ is a triple $(S, \dashrightarrow, \longrightarrow)$, where S is a set of states and $\longrightarrow \subseteq \dashrightarrow \subseteq S \times \Sigma \times S$ are must and may transition relations, respectively.

The MTS are often drawn as follows. Unbroken arrows denote the must (and underlying may) transitions while dashed arrows denote may transitions where there is no must transition.

Example 2.2. The MTS on the right is adapted from [BKL⁺11] and models traffic lights of types used e.g. in Europe and in North America. In state *green* on the left there is a must transition under *ready* to state *yellow* from which there is must transition to *red*. Here transitions to *yellowRed* and back to *green* are may transition. Intuitively, this means that any final implementation may have one or the other transition or both or none. In contrast, the must transitions are present in all implementations.

Note that using MTS, we cannot express the set of implementations with exactly one of the transitions in *red*. For that, we can use Boolean MTS [BKL⁺11] instead, which can express not only arbitrary conjunctions and disjunctions, but also negations and thus also exclusive-or. However, in Boolean MTS it may still happen that at first only transition to *green* is present, but in the next round of the traffic lights cycle only the transition to *yellowRed* is present. To make sure the choice will



remain the same in the whole implementation, parametric MTS have been introduced [BKL⁺11] extending the Boolean MTS.

Before we define the most general class of parametric MTS and derive other classes as special cases, we first recall the standard propositional logic. A Boolean formula over a set X of atomic propositions is given by the following abstract syntax

$$\varphi ::= \mathbf{tt} \mid x \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

where x ranges over X . The set of all Boolean formulae over the set X is denoted by $\mathcal{B}(X)$. Let $\nu \subseteq X$ be a valuation, i.e. a set of variables with value true, then the satisfaction relation $\nu \models \varphi$ is given by $\nu \models \mathbf{tt}$, $\nu \models x$ iff $x \in \nu$, and the satisfaction of the remaining Boolean connectives is defined in the standard way. We also use the standard derived operators like exclusive-or $\varphi \oplus \psi := (\varphi \wedge \neg\psi) \vee (\neg\varphi \wedge \psi)$, implication $\varphi \Rightarrow \psi := \neg\varphi \vee \psi$ and equivalence $\varphi \Leftrightarrow \psi := (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)$.

We can now proceed with the definition of parametric MTS. In essence, it is a labelled transition system where we can specify which transitions can be present depending on values of some fixed parameters.

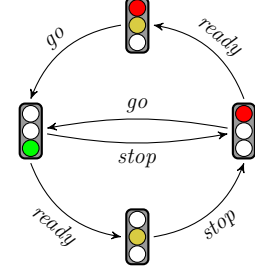
Definition 2.3. A parametric modal transition system (PMTS) over an action alphabet Σ is a tuple (S, T, P, Φ) where

- S is a set of states,
- $T \subseteq S \times \Sigma \times S$ is a transition relation,
- P is a finite set of parameters, and
- $\Phi : S \rightarrow \mathcal{B}((\Sigma \times S) \cup P)$ is an obligation function over the outgoing transitions and parameters. We assume that whenever (a, t) occurs in $\Phi(s)$ then $(s, a, t) \in T$.

A Boolean modal transition system (BMTS) is a PMTS with the set of parameters P being empty. A disjunctive MTS (DMTS) is a BMTS with the obligation function in conjunctive normal form and using no negation. An implementation (or labelled transition system) is a BMTS with $\Phi(s) = \bigwedge_{(s,a,t) \in T} (a, t)$ for each $s \in S$.

An MTS is then a BMTS with $\Phi(s)$ being a conjunction of positive literals (some of the outgoing transitions), for each $s \in S$. More precisely, $--\rightarrow$ is the same as T , and $(s, a, t) \in --\rightarrow$ if and only if (a, t) is one of the conjuncts of $\Phi(s)$.

Example 2.4. An example of a PMTS which captures the traffic lights used e.g. in Europe for cars and for pedestrians is depicted below. Depending on the valuation of parameter *reqYellow*, we either always use the yellow light between the red and green lights, or we never do. The transition relation is depicted using unbroken arrows.



Parameters: $P = \{reqYellow\}$

Obligation function:

$$\begin{aligned}\Phi(green) &= ((stop, red) \oplus (ready, yellow)) \\ &\quad \wedge (reqYellow \Leftrightarrow (ready, yellow)) \\ \Phi(yellow) &= (stop, red) \\ \Phi(red) &= ((go, green) \oplus (ready, yellowRed)) \\ &\quad \wedge (reqYellow \Leftrightarrow (ready, yellowRed)) \\ \Phi(yellowRed) &= (go, green)\end{aligned}$$

3 Modal Refinement

A fundamental advantage of MTS-based formalisms is the presence of *modal refinement* that allows for a step-wise system design (see e.g. [AHL⁺08]). We start with the standard definition of modal refinement for MTS and then discuss extensions to BMTS and PMTS.

Definition 3.1 (MTS Modal Refinement). *For states s_0 and t_0 of MTS $(S_1, \longrightarrow_1, \dashrightarrow_1)$ and $(S_2, \longrightarrow_2, \dashrightarrow_2)$, respectively, we say that s_0 modally refines t_0 , written $s_0 \leq_m t_0$, if (s_0, t_0) is contained in a relation $R \subseteq S_1 \times S_2$ satisfying for every $(s, t) \in R$ and every $a \in \Sigma$:*

1. *if $s \dashrightarrow_1^a s'$ then there is a transition $t \dashrightarrow_2^a t'$ with $(s', t') \in R$, and*
2. *if $t \longrightarrow_2^a t'$ then there is a transition $s \longrightarrow_1^a s'$ with $(s', t') \in R$.*

Intuitively, $s \leq_m t$ iff whatever s can do is allowed by t and whatever t requires can be done by s . Thus s is a refinement of t , or t is an abstraction of s . Further, an *implementation* of s is a state of an implementation (labelled transition system) with $i \leq_m s$.

In [BKL⁺11], the modal refinement has been extended to PMTS (and thus BMTS) so that it coincides on MTS. We first recall the definition for BMTS. To this end, we set the following notation. Let (S, T, P, Φ) be a PMTS and $\nu \subseteq P$ be a valuation. For $s \in S$, we write $T(s) = \{(a, t) \mid (s, a, t) \in T\}$ and denote by

$$\text{Tran}_\nu(s) = \{E \subseteq T(s) \mid E \cup \nu \models \Phi(s)\}$$

the set of all admissible sets of transitions from s under the fixed truth values of the parameters. In the case of BMTS, we often write Tran instead of Tran_\emptyset .

Definition 3.2 (BMTS Modal Refinement). *For states s_0 and t_0 of BMTS $(S_1, T_1, \emptyset, \Phi_1)$ and $(S_2, T_2, \emptyset, \Phi_2)$, respectively, we say that s_0 modally refines t_0 , written $s_0 \leq_m t_0$, if (s_0, t_0) is contained in a relation $R \subseteq S_1 \times S_2$ satisfying for every $(s, t) \in R$:*

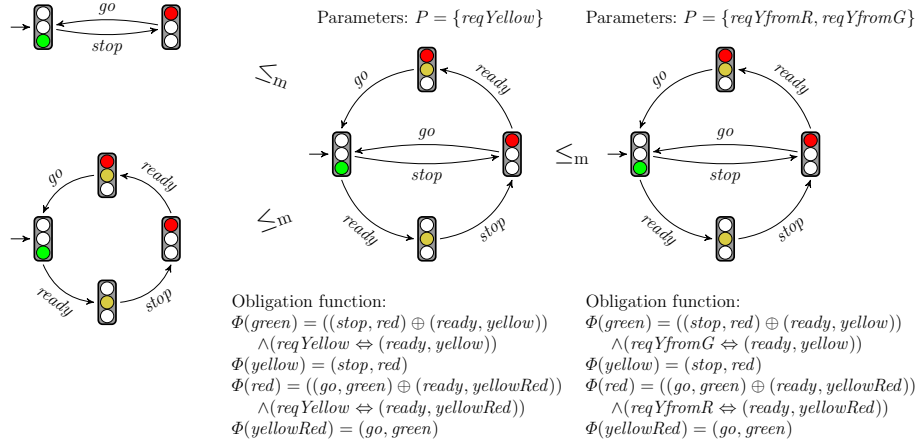
$$\begin{aligned}\forall M \in \text{Tran}(s) : \exists N \in \text{Tran}(t) : & \forall (a, s') \in M : \exists (a, t') \in N : (s', t') \in R \wedge \\ & \forall (a, t') \in N : \exists (a, s') \in M : (s', t') \in R .\end{aligned}$$

For PMTS, we propose here a slightly altered definition, which corresponds more to the intuition, is closer to the semantically defined notion of thorough refinement, but still keeps the same complexity as established in [BKL⁺11]. We use the following notation. For a PMTS $\mathcal{M} = (S, T, P, \Phi)$, a valuation $\nu \subseteq P$ of parameters induces a BMTS $\mathcal{M}^\nu = (S, T, \emptyset, \Phi')$ where each occurrence of $p \in \nu$ in Φ is replaced by **tt** and of $p \notin \nu$ by $\neg\mathbf{tt}$, i.e. $\Phi'(s) = \Phi(s)[\mathbf{tt}/p \text{ for } p \in \nu, \mathbf{ff}/p \text{ for } p \notin \nu]$ for each $s \in S$. We extend the notation to states and let s^ν denote the state of \mathcal{M}^ν corresponding to the state s of \mathcal{M} .

Definition 3.3 (PMTS Modal Refinement). *For states s_0 and t_0 of PMTS (S_1, T_1, P_1, Φ_1) and (S_2, T_2, P_2, Φ_2) , we say that s_0 modally refines t_0 , written $s_0 \leq_m t_0$, if for every $\mu \subseteq P_1$ there exists $\nu \subseteq P_2$ such that $s_0^\mu \leq_m t_0^\nu$.*

Before we comment on the difference to the original definition, we illustrate the refinement on an example of [BKL⁺11] where both definitions coincide.

Example 3.4. Consider the rightmost PMTS below. It has two parameters, namely *reqYfromG* and *reqYfromR* whose values can be set independently and it can be refined by the system in the middle of the figure having only one parameter *reqYellow*. This single parameter simply binds the two original parameters to the same value. The PMTS in the middle can be further refined into the implementations where either yellow is always used in both cases, or never at all as discussed in the previous example. Up to bisimilarity, the *green* state of this system only has the two implementations on the left.

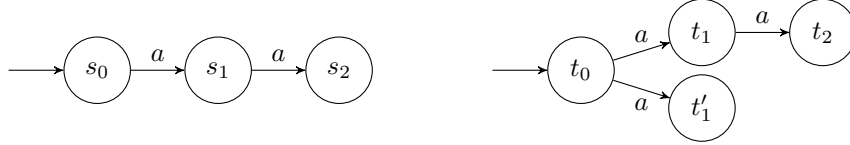


The original version of [BKL⁺11] requires for $s_0 \leq_m t_0$ to hold that there be a fixed $R \subseteq S_1 \times S_2$ such that for every $\mu \subseteq P_1$ there exists $\nu \subseteq P_2$ satisfying for each $(s, t) \in R$

$$\forall M \in \text{Tran}_\mu(s) : \exists N \in \text{Tran}_\nu(t) : \forall (a, s') \in M : \exists (a, t') \in N : (s', t') \in R \wedge \forall (a, t') \in N : \exists (a, s') \in M : (s', t') \in R.$$

Clearly, the original definition is stronger: For any two PMTS states, if $s_0 \leq_m t_0$ holds according to [BKL⁺11] it also holds according to Definition 3.3. Indeed, the relation for any sets of parameters can be chosen to be the fixed relation R . On the other hand, the opposite does not hold.

Example 3.5. Consider the PMTS on the left with parameter set $\{p\}$ and obligation $\Phi(s_0) = (a, s_1)$, $\Phi(s_1) = (b, s_2) \Leftrightarrow p$, $\Phi(s_2) = \mathbf{tt}$ and the PMTS on the right with parameter set $\{q\}$ and obligation $\Phi(t_0) = ((a, t_1) \Leftrightarrow q) \wedge ((a, t'_1) \Leftrightarrow \neg q)$, $\Phi(t_1) = (a, t_2)$, $\Phi(t_2) = \Phi(t'_1) = \mathbf{tt}$. On the one hand, according to our definition $s_0 \leq_m t_0$. We intuitively agree it should be the case (and note they also have the same set of implementations). On the other hand, the original definition does not allow to conclude modal refinement between s_0 and t_0 . The reason is that depending on the value of p , s_1 is put in the relation either with t_1 (for p being true and thus choosing q true, too) or with t'_1 (for p being false and thus choosing q false, too). In contrast to the original definition, our definition allows us to pick different relations for different parameter valuations.



We propose our modification of the definition since it is more intuitive and for all considered fragments of PMTS has the same complexity as the original one. Note that both definitions coincide on BMTS. Further, on MTS they coincide with Definition 3.1 and on labelled transition systems with bisimulation.

4 Modal Refinement Checking

In this section, we show how to solve the modal refinement problem on BMTS and PMTS using QBF solvers. Although modal refinement is Π_2 -complete (the second level of the polynomial hierarchy) on BMTS and Π_4 -complete on PMTS (see [BKL⁺11]), this way we obtain a solution method that is practically fast. We have implemented the approach and document its scalability on experimental results.

As mentioned, in order to decide whether modal refinement holds between two states, a reduction to a quantified boolean formula will be used. First, we recall the QBF decision problems.

Definition 4.1 (QBF_n^Q). *Let Ap be a set of atomic propositions, which is partitioned into n sets with $Ap = \bigcup_{i=0}^n X_i$, and $\phi \in \mathcal{B}(Ap)$ a boolean formula over this set of atomic propositions. Let $Q \in \{\forall, \exists\}$ be a quantifier and $\bar{\cdot} : \{\forall \mapsto \exists, \exists \mapsto \forall\}$ a function. Then a formula*

$$QX_1 \bar{Q}X_2 QX_3 \dots \bar{Q}X_n \phi \quad \text{with } \bar{Q} = \begin{cases} Q & \text{if } n \text{ is odd} \\ \bar{Q} & \text{if } n \text{ is even} \end{cases}$$

is an instance of QBF_n^Q if it is satisfiable.

Satisfiability means that if e.g. $Q = \exists$ there is some partial valuation for the atomic propositions in X_1 , such that for all partial valuations for the elements of X_2 , there is another partial valuation for the propositions of X_3 and so on up to X_n , such that ϕ is satisfied by the union of all partial valuations. It is well known that these problems are complete for the polynomial hierarchy: For each $i \geq 1$, QBF_i^\exists is Σ_i -complete and QBF_i^\forall is Π_i -complete.

4.1 Construction for BMTS

Due to the completeness of QBF problems and the results of [BKL⁺11], it is possible to polynomially reduce modal refinement on BMTS to QBF_2^\forall . However, we would then have to perform a fixpoint computation to compute the refinement relation causing numerous invocations of the external QBF solver. Hence it is faster to guess the relation and thus reduce the modal refinement only to QBF_3^\exists .

Let $s \in S_1$ and $t \in S_2$ be processes of two arbitrary BMTSs $\mathcal{M}_1 = (S_1, T_1, \emptyset, \Phi_1)$ and $\mathcal{M}_2 = (S_2, T_2, \emptyset, \Phi_2)$. Furthermore let

$$Ap = \underbrace{(S_1 \times S_2)}_{X_R} \uplus \underbrace{T_1}_{X_{T1}} \uplus \underbrace{(S_1 \times T_2)}_{X_{T2}}$$

be a set of atomic propositions. The intended meaning is that $(u, v) \in X_R$ is assigned **tt** if and only if it is also contained in the modal refinement relation R . Further, X_{T1} and X_{T2} are used to talk about the transitions. The prefix S_1 is attached to the set T_2 because $N \in \text{Tran}(t)$ with $t \in S_2$ must be chosen independently for different states of S_1 . This trick enables us later to pull up the \exists quantification in the formula.

We now construct a formula $\Psi_{s,t} \in \mathcal{B}(Ap)$ satisfying

$$s \leq_m t \quad \text{iff} \quad \exists X_R \forall X_{T1} \exists X_{T2} \Psi_{s,t} \in QBF_3^\exists \quad (1)$$

To this end, we shall use a macro $\psi_{u,v}$ capturing the condition which has to be satisfied by any element $(u, v) \in R$. Furthermore, we ensure that (s, t) is assigned **tt** by every satisfying assignment for the formula by placing it directly in the conjunction:

$$\Psi_{s,t} = (s, t) \wedge \bigwedge_{(u,v) \in X_R} ((u, v) \Rightarrow \psi_{u,v}) \quad (2)$$

It remains to define the macro $\psi_{u,v}$. We start with the modal refinement condition as a blueprint:

$$\forall M \in \text{Tran}(u) : \exists N \in \text{Tran}(v) : \forall (a, u') \in M : \exists (a, v') \in N : (u', v') \in R \wedge \\ \forall (a, v') \in N : \exists (a, u') \in M : (u', v') \in R .$$

As M and N are subsets of $T_1(u)$ and $T_2(v)$, respectively, and are finite, the inner quantifiers can be expanded causing only a polynomial growth of the

formula size (see Appendix A). Further, Tran sets are replaced by the original definition and the outer quantifiers are moved in front of $\Psi_{s,t}$. As the state obligations are defined over a different set of atomic propositions ($\Phi(v) \in \mathcal{B}((\Sigma \times S) \cup P) \not\subseteq \mathcal{B}(Ap)$), a family of mapping functions π_p is introduced.

$$\begin{aligned}
\pi_p : \mathcal{B}(\Sigma \times S) &\rightarrow \mathcal{B}(Ap) \\
\mathbf{tt} &\mapsto \mathbf{tt} \\
(a, x) &\mapsto (p, a, x) \quad \text{with } a \in \Sigma, x \in S \\
\neg\varphi &\mapsto \neg\pi_p(\varphi) \\
\varphi_1 \wedge \varphi_2 &\mapsto \pi_p(\varphi_1) \wedge \pi_p(\varphi_2) \\
\varphi_1 \vee \varphi_2 &\mapsto \pi_p(\varphi_1) \vee \pi_p(\varphi_2)
\end{aligned} \tag{3}$$

Applying these steps to the blueprint yields the following result:

$$\psi_{u,v} = \pi_u(\Phi_1(u)) \Rightarrow \pi_{u,v}(\Phi_2(v)) \wedge \varphi_{u,v} \tag{4}$$

$$\begin{aligned}
\varphi_{u,v} &= \bigwedge_{\substack{u^* \in X_{T1} \\ u^* = (u, a, u')}} (u^* \Rightarrow \bigvee_{\substack{v^* \in X_{T2} \\ v^* = (u, v, a, v')}} (v^* \wedge (u', v'))) \\
&\quad \wedge \bigwedge_{\substack{v^* \in X_{T2} \\ v^* = (u, v, a, v')}} (v^* \Rightarrow \bigvee_{\substack{u^* \in X_{T1} \\ u^* = (u, a, u')}} (u^* \wedge (u', v')))
\end{aligned} \tag{5}$$

Theorem 4.2. *For states s, t of a BMTS, we have*

$$s \leq_m t \quad \text{iff} \quad \exists X_R \forall X_{T1} \exists X_{T2} \Psi_{s,t} \in QBF_3^\exists$$

Due to space constraints, the technical proof is moved to Appendix A.

4.2 Construction for PMTS

We now reduce the modal refinement on PMTS to QBF_4^\forall , which now corresponds directly to the complexity established in [BKL⁺11]. Nevertheless, due to the first existential quantification in $\forall\exists\forall\exists$ alternation sequence, we can still guess the refinement relation using the QBF solver rather than compute the lengthy fixpoint computation.

In the PMTS case, we have to find for all parameter valuations for the system of s a valuation for the system of t , such that there exists a modal refinement relation containing (s, t) . We simply choose universally a valuation for the parameters of the left system (the underlying system of s) and then existentially for the right system (the underlying system of t). Prior to checking modal refinement, the valuations are fixed, so the PMTS becomes a BMTS. This is accomplished by extending Ap with P_1 and P_2 and adding the necessary quantifiers to the formula. Thus we obtain the following:

Theorem 4.3. *For states s, t of a PMTS, we have*

$$s \leq_m t \quad \text{iff} \quad \forall P_1 \exists P_2 \exists X_R \forall X_{T1} \exists X_{T2} \Psi_{s,t} \in QBF_4^\forall$$

4.3 Experimental Results

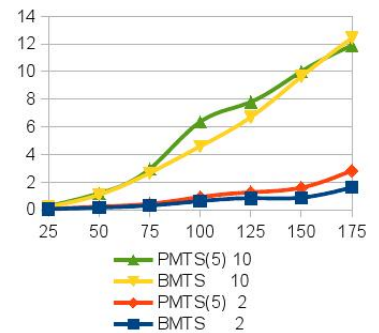
We now show how our method performs in practice. We implemented the reduction and linked it to the QBF solver Quantor. In order to evaluate whether our solution scales, we generate random samples of MTS, disjunctive MTS, Boolean MTS and parametric MTS with different numbers of parameters (as displayed in tables below in parenthesis). For each type of system and the number of reachable states (25 to 200 as displayed in columns), we generate several pairs of systems and compute the average time to check modal refinement between them.

We show several sets of experiments. In Table 1, we consider (1) systems with alphabet of size 2 and all states with branching degree 2, and (2) systems with alphabet of size 10 and all states with branching degree 10. Further, in Table 2, we consider systems with alphabet of size 2 and all states with branching degree 5. Here we first consider the systems as above, i.e. with edges generated randomly so that they create a tree and with some additional “noise” edges thus making the branching degree constant. Second, we consider systems where we have different “clusters”, each of which is interconnected with many edges. Each of these clusters has a couple of “interface” states, which are used to connect to other clusters. We use this class of systems to model system descriptions with more organic structure.

The entries in the tables are average running times in seconds. The standard deviation in our experiments was around 30-60%. Each star denotes that on one of five experiments, the QBF solver Quantor timed out after one minute. The experiments were run on Intel Core 2 Duo CPU P9600 2.66GHz x 2 with 3.8 GB RAM using Java 1.7. For more details and more experiments, see <http://www.model.in.tum.de/~kretinsk/ictac13.html>.

Table 1. Experimental results: systems over alphabet of size 2 with branching degree 2 in the upper part, and systems over alphabet of size 10 with branching degree 10 in the lower part

	25	50	75	100	125	150	175	200
MTS	0.03	0.15	0.29	0.86	0.87	0.96	1.88	2.48
DMTS	0.04	0.22	0.39	0.91	1.13	1.34	2.61	3.19
BMTS	0.03	0.15	0.30	0.62	0.83	0.87	1.61	2.17
PMTS(1)	0.03	0.20	0.37	0.84	0.97	1.23	2.44	3.15
PMTS(5)	0.04	0.22	0.42	0.91	1.26	1.59	2.83	3.66
MTS	0.18	0.84	2.12	3.88	5.63	7.64	10.30	14.18
DMTS	0.44	2.23	5.31	8.59	10.13	14.14	13.96	66.92
BMTS	0.21	1.08	2.65	4.58	6.70	9.63	12.44	17.06
PMTS(1)	0.26	1.12	2.74	4.57	7.58	10.31	11.26	16.41
PMTS(5)	0.25	1.17	2.94	6.36	7.80	10.01	11.90	36.51



On the one hand, observe that the number of parameters does not play any major rôle in the running time. The running times on PMTS with 5 or even more parameters are very close to BMTS, i.e. PMTS with zero parameters, as can be seen in the graph. Therefore, the greatest theoretical complexity threat—the

Table 2. Experimental results: systems over alphabet of size 2 with branching degree 5; systems with random structure in the upper part, and systems with organic structure in the lower part

	25	50	75	100	125	150	175	200
PMTS (1)	0.34	2.04	5.38	8.81	11.78	17.41	27.33	58.06
PMTS (5)	0.29	1.83	*5.19	12.79	15.71	26.60	*35.30	89.25
PMTS (10)	*0.43	1.36	6.70	13.66	*18.27	*21.10	51.67	232.83
PMTS (1)	0.05	0.14	0.18	0.30	3.40	0.73	0.85	0.96
PMTS (5)	0.02	0.04	0.23	0.70	0.58	0.39	1.13	*2.35
PMTS (10)	0.02	0.10	0.16	*0.16	*0.29	1.55	0.97	1.13

number of parameters allowing in general only for searching all exponentially many combinations—is in practice eliminated by the use of QBF solvers.

On the other hand, observe that the running time is more affected by the level of non-determinism. For branching degree 10 over 10-letter alphabet there, there are more likely to be more outgoing transitions under the same letter than in the case with branching degree 2 over 2-letter alphabet, but still less than for branching degree 5 over 2-letter alphabet. However, the level of non-determinism is often quite low [BKLS09], hence this dependency does not pose so serious problem in practice. Further, even this most difficult setting with high level of non-determinism allows for fast analysis if systems with natural organic structure are considered, cf. upper and lower part of Table 2.

A more serious problem stems from our use of Java. With sizes around 200, the running times often get considerably longer, see the tables. Here the memory management and the garbage collection take their toll. However, this problem should diminish in a garbage-collection-free setting.

5 Thorough Refinement Checking

While modal refinement has been defined syntactically, there is also a corresponding notion defined semantically. The semantics of a state s of a PMTS is the set of its implementations $\llbracket s \rrbracket := \{i \mid i \text{ is an implementation and } i \leq_m s\}$.

Definition 5.1 (Thorough Refinement). *For states s_0 and t_0 of PMTS, we say that s_0 thoroughly refines t_0 , written $s_0 \leq_t t_0$, if $\llbracket s_0 \rrbracket \subseteq \llbracket t_0 \rrbracket$.*

5.1 Transforming PMTS to BMTS and DMTS

The thorough refinement problem is EXPTIME-complete for MTS [BKLS12] and also for DMTS [BČK11] (for proof, see [BČK10]). First, we show how to transform PMTS to BMTS and DMTS and thus reduce our problems to the already solved one.

For a PMTS, we define a system where we can use any valuation of the parameters:

Definition 5.2. For a PMTS $\mathcal{M} = (S, T, P, \Phi)$ with initial state s_0 , we define a BMTS called de-parameterization $\mathcal{M}^B = (\{s_0^B\} \cup S \times 2^P, T', \emptyset, \Phi')$ with initial state s_0^B and

- $T = \{(s_0^B, a, (s, \nu)) \mid (s_0, a, s) \in T, \nu \subseteq P\} \cup \{((s, \nu), a, (s', \nu)) \mid (s, a, s') \in T\},$
- $\Phi'(s_0^B) = \bigoplus_{\nu \subseteq P} \Phi(s_0)[\mathbf{tt}/p \text{ for } p \in \nu, \mathbf{ff}/p \text{ for } p \notin \nu, (s, \nu)/s],$
- $\Phi'((s, \nu)) = \Phi(s)[\mathbf{tt}/p \text{ for } p \in \nu, \mathbf{ff}/p \text{ for } p \notin \nu, (s, \nu)/s].$

The de-parameterization is a BMTS having exactly all the implementations of the PMTS and only one (trivial) valuation.

Proposition 5.3. Let s_0 be a PMTS state. Then $\llbracket s_0 \rrbracket = \llbracket s_0^B \rrbracket$ and $s_0 \leq_m s_0^B$.

Proof. For any parameter valuation ν we match it with \emptyset and the modal refinement is achieved in the copy with ν fixed in the second component. Clearly, any implementation of s_0^B corresponds to a particular parameter valuation and thus also to an implementation of s_0 . \square

Remark 5.4. The price we have to pay is a blowup exponential in $|P|$. This is, however, inevitable. Indeed, consider a PMTS $(\{s_0, s_1, s_2\}, \{(s_0, p, s_1), (s_1, p, s_2) \mid p \in P\}, P, \{s_0, s_1 \mapsto \bigwedge_{p \in P} (p, s) \Leftrightarrow p, s_2 \mapsto \mathbf{tt}\})$. Then in every equivalent BMTS we need to remember the transitions of the first step so that we can repeat exactly these in the following step. Since there are exponentially many possibilities, the result follows.

Further, similarly to Boolean formulae with states in $[\text{BDF}^+]$, we can transform every BMTS to a DMTS.

Definition 5.5. For a BMTS $\mathcal{M} = (S, T, \emptyset, \Phi)$ with initial state s_0 , we define a DMTS called de-negation $\mathcal{M}^D = (S', T', \emptyset, \Phi')$

- $S' = \{M \in \text{Tran}(s) \mid s \in S\},$
- $\Phi'(M) = \bigwedge_{(a, s') \in M} \bigvee_{M' \in \text{Tran}(s')} (a, M'),$

and T' minimal such that for each $M \in S'$ and each occurrence of (a, M') in $\Phi(M)$, we have $(M, a, M') \in T'$.

However, this DMTS needs to have more initial states in order to be equivalent to the original BMTS:

Lemma 5.6. For a state s_0 of a BMTS, $\llbracket s_0 \rrbracket = \bigcup_{M \in \text{Tran}(s_0)} \llbracket M \rrbracket$ (where M are taken as states of the de-negation).

Note that both transformations are exponential. The first one in $|P|$ and the second one in the branching degree. Therefore, their composition is still only singly exponential yielding a state space where each state has two components: a valuation of original parameters and Tran of the original state under this valuation.

Theorem 5.7. *Thorough refinement on PMTS is in 2-EXPTIME.*

Proof. Recall that thorough refinement on DMTS is in EXPTIME. Further, note that we have reduced the PMTS and BMTS thorough refinement problems to the one on DMTS with more initial states. However, this does not pose a problem. Indeed, let s_0 and t_0 be states of a BMTS. We want to check whether $s_0 \leq_t t_0$. According to [BČK10] where DMTS only have one initial state, we only need to check whether for each $M \in \text{Tran}(s_0)$ we have $(M, \text{Tran}(t_0)) \notin \text{Avoid}$, which can clearly still be done in exponential time. \square

5.2 Direct algorithm

We now extend the approach for MTS and DMTS to the BMTS case. Before proceeding, one needs to prune all inconsistent states, i.e. those with unsatisfiable obligation. This is standard and the details can be found in Appendix B.

We define a set *Avoid*, which contains pairs consisting of one process and one set of processes. A pair is contained in the relation if there exists an implementation refining the single process, but none of the other processes. This approach is very similar to [BKLS12], but the rules for generating *Avoid* are much more complex.

Definition 5.8. (Avoid) *Let (S, T, \emptyset, Φ) be a globally consistent BMTS over the action alphabet Σ . The set of avoiding states of the form (s, \mathcal{T}) , where $s \in S$ and $\mathcal{T} \subseteq S$, is the smallest set *Avoid* such that $(s, \mathcal{T}) \in \text{Avoid}$ whenever $\mathcal{T} = \emptyset$ or there exists an admissible set of transitions $M \in \text{Tran}(s)$ and sets $\text{later}_{a,u,f} \subseteq S$ for every $a \in \Sigma$, $u \in S$, $f \in \bigcup_{t \in \mathcal{T}} \text{Tran}(t)$ such that*

$$\begin{aligned} & \forall t \in \mathcal{T} : \forall N_t \in \text{Tran}(t) : \exists a \in \Sigma : \\ & \quad \exists t_a \in N_t(a) : \forall s_a \in M(a) : \forall f \in \bigcup_{t \in \mathcal{T}} \text{Tran}(t) : t_a \in \text{later}_{a,s_a,f} \\ & \vee \quad \exists s_a \in M(a) : \forall t_a \in N_t(a) : t_a \in \text{later}_{a,s_a,N_t} \end{aligned}$$

and

$$\forall f \in \bigcup_{t \in \mathcal{T}} \text{Tran}(t) : \forall (a, s_a) \in M : (s_a, \text{later}_{a,s_a,f}) \in \text{Avoid}$$

hold.

Lemma 5.9. *Given processes $s, t_1, t_2 \dots t_n$ of some finite, global-consistent BMTS, there exists an implementation I such that $I \leq_m s$ and $I \not\leq_m t_i$ for all $i \in [1, n]$ if $(s, \{t_1, t_2 \dots t_n\}) \in \text{Avoid}$.*

Theorem 5.10. *Thorough refinement checking on BMTS is in NEXPTIME.*

Proof. For deciding $s \leq_t t$ the *Avoid* relation has to be computed, whose size grows exponentially with the size of the underlying system. Moreover, in each step of adding a new element is added to *Avoid*, the sets $\text{later}_{a,s,f}$ need to be guessed. \square

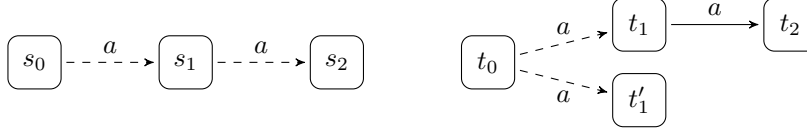
6 Thorough vs. Modal Refinement

In this section, we discuss the relationship of the two refinements. Some proofs are moved to Appendix C. Firstly, the modal refinement is a sound approximation to the thorough refinement.

Proposition 6.1. *Let s_0 and t_0 be states of PMTS. If $s_0 \leq_m t_0$ then also $s_0 \leq_t t_0$.*

Proof. For any $i \in \llbracket s_0 \rrbracket$, we have $i \leq_m s_0$ and due to transitivity of \leq_m , $i \leq_m s_0 \leq_m t_0$ implies $i \leq_m t_0$, hence $i \in \llbracket t_0 \rrbracket$. \square

The converse fails already for MTS as shown in the following classical example ([BKLS09]) where $s_0 \leq_t t_0$, but $s_0 \not\leq_m t_0$.

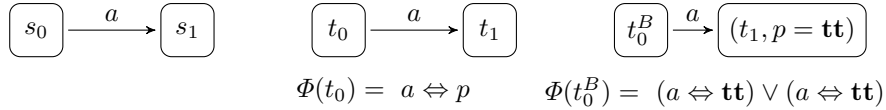


However, provided the refined MTS is deterministic, the approximation is also complete [BKLS09]. This holds also for BMTS. This is very useful as deterministic system often appear in practice [BKLS09] and checking modal refinement is computationally easier than the thorough refinement. Formally, we say that a PMTS (S, T, P, Φ) is deterministic if for every $(s, a, t), (s, a, t') \in T$ we have $t = t'$.

Proposition 6.2. *Let s_0 be a PMTS state and t_0 a deterministic BMTS state. If $s_0 \leq_t t_0$ then also $s_0 \leq_m t_0$.*

However, the completeness fails if the refined system is deterministic but with parameters:

Example 6.3. Consider a BMTS $(\{s_0, s_1\}, \{s_0, a, s_1\}, \emptyset, \{s_0 \mapsto \mathbf{tt}, s_1 \mapsto \mathbf{tt}\})$ and a deterministic PMTS $(\{t_0, t_1\}, \{(t_0, a, t_1)\}, \{p\}, \{t_0 \mapsto a \Leftrightarrow p, t \mapsto \mathbf{tt}\})$ below. Obviously $\llbracket s_0 \rrbracket = \llbracket t_0 \rrbracket$ contains the implementations with no transitions or one step a -transitions. Although $s_0 \leq_t t_0$, we do not have $s_0 \leq_m t_0$ as we cannot match with any valuation of p .



Corollary 6.4. *There is a state s_0 of a PMTS and a state t_0 of a deterministic PMTS such that $s_0 \leq_t t_0$ but $s_0 \not\leq_m t_0$.*

In the previous example, we lacked the option to match a system with different parameter valuations at once. However, the de-parameterization introduced earlier is non-deterministic even if the original system was deterministic. Hence

the modal refinement is not guaranteed to coincide with the thorough refinement. In [BKLS09], we defined the notion of deterministic hull, the best deterministic overapproximation of a system. The construction on may transitions was the standard powerset construction and a must transition was created if all states of a macrostate had one. Here we extend this notion to PMTS, which allows to over- and under-approximate the thorough refinement by the modal refinement.

Definition 6.5. For a PMTS $\mathcal{M} = (S, T, P, \Phi)$ with initial state s_0 , we define a PMTS called deterministic hull $\mathcal{D}(\mathcal{M}) = (2^S, T', P, \Phi')$ with initial state $\mathcal{D}(s_0) := \{s_0\}$ and

- $T = \{(S, a, S_a)\}$ where S_a denotes all a -successors of elements of S , i.e. $S_a = \{s' \mid \exists s \in S : (s, a, s') \in T\}$,
- $\Phi'(S) = \bigvee_{s \in S} \Phi(s)[(a, S_a)/(a, s)]$ for every a, s .

Proposition 6.6. For a PMTS state s_0 , $\mathcal{D}(s_0)$ is deterministic and $s_0 \leq_m \mathcal{D}(s_0)$.

We now show the minimality of the deterministic hull.

Proposition 6.7. Let s_0 be a PMTS state. Then

- for every deterministic PMTS state t_0 , if $s_0 \leq_m t_0$ then $\mathcal{D}(s_0) \leq_m t_0$;
- for every deterministic BMTS state t_0 , if $s_0 \leq_t t_0$ then $\mathcal{D}(s_0) \leq_m t_0$.

The next transformation allows for removing the parameters without introducing non-determinism.

Definition 6.8. For a PMTS $\mathcal{M} = (S, T, P, \Phi)$ with initial state s_0 , we define a BMTS called parameter-free hull $\mathcal{P}(\mathcal{M}) = (S, T, \emptyset, \Phi')$ with initial state $\mathcal{P}(s_0) := s_0$ and

$$\Phi'(s) = \bigvee_{\nu \subseteq P} \Phi(s)[\mathbf{tt}/p \text{ for } p \in \nu, \mathbf{ff}/p \text{ for } p \notin \nu]$$

Lemma 6.9. For a PMTS state s_0 , $s_0 \leq_m s_0^B \leq_m \mathcal{P}(s_0)$.

The parameter-free deterministic hull now plays the rôle of the deterministic hull for MTS.

Corollary 6.10. For PMTS states s_0 and t_0 , if $s_0 \leq_t t_0$ then $s_0 \leq_m \mathcal{P}(\mathcal{D}(t_0))$.

Proof. Since $s_0 \leq_t t_0$, we also have $s_0 \leq_t \mathcal{D}(t_0)$ by Propositions 6.6 and 6.1. Therefore, $s_0 \leq_t \mathcal{P}(\mathcal{D}(t_0))$ by Proposition 6.9 and thus $s_0 \leq_m \mathcal{P}(\mathcal{D}(t_0))$ by Proposition 6.2. \square

7 Conclusions

We have investigated both modal and thorough refinement on Boolean and parametric extension of modal transition systems. Apart from results summarized in the table below, we have shown a practical way to compute modal refinement and use it for approximating thorough refinement. Closing the complexity gap for thorough refinement, i.e. obtaining matching lower bounds or improving our algorithm remains as an open question.

	MTS	BMTS	PMTS
$\leq_t \in$	EXPTIME	NEXPTIME	2-EXPTIME
refined system deterministic	$\leq_m = \leq_t$	$\leq_m = \leq_t$	$\leq_m \neq \leq_t$

References

- AFdFE⁺11. L. Aceto, I. Fábregas, D. de Frutos-Escrig, A. Ingólfssdóttir, and M. Palomino. Graphical representation of covariant-contravariant modal formulae. In *EXPRESS*, pages 1–15, 2011.
- AHKV98. R. Alur, T. A. Henzinger, O. Kupferman, and M. Y. Vardi. Alternating refinement relations. In *CONCUR*, pages 163–178, 1998.
- AHL⁺08. A. Antonik, M. Huth, K. G. Larsen, U. Nyman, and A. Wasowski. 20 years of modal and mixed specifications. *Bulletin of the EATCS no. 95*, pages 94–129, 2008.
- BČK10. N. Beneš, I. Černá, and J. Křetínský. Disjunctive modal transition systems and generalized LTL model checking. Technical report FIMU-RS-2010-12, Faculty of Informatics, Masaryk University, Brno, 2010.
- BČK11. N. Beneš, I. Černá, and J. Křetínský. Modal transition systems: Composition and LTL model checking. In *ATVA*, pages 228–242, 2011.
- BDF⁺. N. Beneš, B. Delahaye, U. Fahrenberg, J. Křetínský, and A. Legay. Hennessy-milner logic with maximal fixed points as a specification theory. Submitted.
- BKL⁺11. N. Beneš, J. Křetínský, K. G. Larsen, M. H. Møller, and J. Srba. Parametric modal transition systems. In *ATVA*, pages 275–289, 2011.
- BKL⁺12. N. Beneš, J. Křetínský, K. G. Larsen, M. H. Møller, and J. Srba. Dual-priced modal transition systems with time durations. In *LPAR*, pages 122–137, 2012.
- BKLS09. N. Beneš, J. Křetínský, K. G. Larsen, and J. Srba. On determinism in modal transition systems. *Theor. Comput. Sci.*, 410(41):4026–4043, 2009.
- BKLS12. N. Beneš, J. Křetínský, K. G. Larsen, and J. Srba. Exptime-completeness of thorough refinement on modal transition systems. *Inf. Comput.*, 218:54–68, 2012.
- BL90. G. Boudol and K. G. Larsen. Graphical versus logical specifications. In *CAAP*, pages 57–71, 1990.
- BLS95. A. Børjesson, K. G. Larsen, and A. Skou. Generality in design and compositional verification using TAV. *Formal Methods in System Design*, 6(3):239–258, 1995.
- BML11. S. S. Bauer, P. Mayer, and A. Legay. MIO workbench: A tool for compositional design with modal input/output interfaces. In *ATVA*, pages 418–421, 2011.
- Bru97. G. Bruns. An industrial application of modal process logic. *Sci. Comput. Program.*, 29(1-2):3–22, 1997.

- CGLT09. A. Campetelli, A. Gruler, M. Leucker, and D. Thoma. *Don't Know* for multi-valued systems. In *ATVA*, pages 289–305, 2009.
- dAH01. L. de Alfaro and T. A. Henzinger. Interface automata. In *ESEC / SIG-SOFT FSE*, pages 109–120, 2001.
- DFFU07. N. D'Ippolito, D. Fischbein, H. Foster, and S. Uchitel. MTSA: Eclipse support for modal transition systems construction, analysis and elaboration. In *ETX*, pages 6–10, 2007.
- DGG97. D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Trans. Program. Lang. Syst.*, 19(2):253–291, 1997.
- DN04. D. Dams and K. S. Namjoshi. The existence of finite abstractions for branching time model checking. In *LICS*, pages 335–344, 2004.
- FS08. H. Fecher and H. Schmidt. Comparing disjunctive modal transition systems with an one-selecting variant. *J. Log. Algebr. Program.*, 77(1-2):20–39, 2008.
- GHJ01. P. Godefroid, M. Huth, and R. Jagadeesan. Abstraction-based model checking using modal transition systems. In *CONCUR*, pages 426–440, 2001.
- GNRT10. P. Godefroid, A. V. Nori, S. K. Rajamani, and S. Tetali. Compositional may-must program analysis: unleashing the power of alternation. In *POPL*, pages 43–56, 2010.
- HJS01. M. Huth, R. Jagadeesan, and D. A. Schmidt. Modal transition systems: A foundation for three-valued program analysis. In *ESOP*, pages 155–169, 2001.
- JP01. B. Jacobs and E. Poll. A logic for the java modeling language JML. In *FASE*, pages 284–299, 2001.
- LNW07. K. G. Larsen, U. Nyman, and A. Wasowski. Modal I/O automata for interface and product line theories. In *ESOP*, pages 64–79, 2007.
- LT88. K. G. Larsen and B. Thomsen. A modal process logic. In *LICS*, pages 203–210, 1988.
- LW94. B. Liskov and J. M. Wing. A behavioral notion of subtyping. *ACM Trans. Program. Lang. Syst.*, 16(6):1811–1841, 1994.
- LX90. K. G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *LICS*, pages 108–117, 1990.
- Lyn88. N. Lynch. I/O automata: A model for discrete event systems. In *22nd Annual Conference on Information Sciences and Systems*, pages 29–38. Princeton University, 1988.
- Nam03. K. S. Namjoshi. Abstraction for branching time properties. In *CAV*, pages 288–300, 2003.
- Nym08. U. Nyman. *Modal Transition Systems as the Basis for Interface Theories and Product Lines*. PhD thesis, Institut for Datalogi, Aalborg Universitet, 2008.
- Rac07. J.-B. Raclet. *Quotient de spécifications pour la réutilisation de composants*. PhD thesis, Université de Rennes I, december 2007. (In French).
- RBB⁺09. J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, and R. Passerone. Why are modalities good for interface theories? In *ACSD*. IEEE Computer Society Press, 2009.
- RBB⁺11. J.-B. Raclet, E. Badouel, A. Benveniste, B. Caillaud, A. Legay, and R. Passerone. A modal interface theory for component-based design. *Fundamenta Informaticae*, 108(1-2):119–149, 2011.
- UC04. S. Uchitel and M. Chechik. Merging partial behavioural models. In *SIG-SOFT FSE*, pages 43–52, 2004.

Appendix: Proofs

A Modal Refinement Checking: Proof of Theorem 4.3

Before proving the soundness and the correctness of the construction for BMTS, a lemma is introduced to simplify this proof.

Lemma A.1. *Let be $(s, t) \in S_1 \times S_2$ a pair of states. Let be \mathcal{A}_{X_R} , $\mathcal{A}_{X_{T_1}}$ and $\mathcal{A}_{X_{T_2}}$ partial valuations for the sets of atomic propositions appearing in their indices. Furthermore let be $R \subseteq S_1 \times S_2$, $M \in \text{Tran}_\emptyset(s)$ and $N \in \text{Tran}_\emptyset(t)$ sets. If $\mathcal{A}_{X_R} = R$, $\mathcal{A}_{X_{T_1}} \supseteq \pi_s(M)$ and $\mathcal{A}_{X_{T_2}} \supseteq \pi_{s,t}(N)$ holds, then $\mathcal{A}_{X_R} \cup \mathcal{A}_{X_{T_1}} \cup \mathcal{A}_{X_{T_2}} \models \varphi_{s,t}$ if and only if*

$$\begin{aligned} R \cup M \cup N \models & \quad \forall(a, s') \in M : \exists(a, t') \in N : (s', t') \in R \\ & \wedge \quad \forall(a, t') \in N : \exists(a, s') \in M : (s', t') \in R \end{aligned}$$

Proof. We assume the conditions and set $\mathcal{A}_X = \mathcal{A}_{X_R} \cup \mathcal{A}_{X_{T_1}} \cup \mathcal{A}_{X_{T_2}}$ and $\mathcal{A}_R = R \cup M \cup N$. Additionally, we only consider one half of the conjunction, as the other is proven analogously.

$$\begin{aligned} \mathcal{A}_R \models & \quad \forall(a, s') \in M : \exists(a, t') \in N : (s', t') \in R \\ \text{iff} \quad \mathcal{A}_R \models & \quad \bigwedge_{(a, s') \in T_1(s)} ((a, s') \in M \Rightarrow \bigvee_{(a, t') \in T_2(t)} ((a, t') \in N \wedge (s', t') \in R)) \\ \text{iff} \quad \mathcal{A}_X \models & \quad \bigwedge_{\substack{s^* \in X_{T_1} \\ s^* = (s, a, s')}} (s^* \Rightarrow \bigvee_{\substack{t^* \in X_{T_2} \\ t^* = (s, t, a, t')}} (t^* \wedge (s', t'))) \end{aligned}$$

As M and N are finite sets, \forall and \exists quantifiers may simply be expanded. In the second step we simply apply π and substitute \in with atomic propositions. \square

A relation satisfying the conditions of the definition of the modal refinement is called a *modal refinement relation*.

Soundness and Correctness 'If' part (soundness of the construction). Assume $s \leq_m t$ with the modal refinement relation R . As the partial valuation for X_R , we set $\mathcal{A}_{X_R} = R$. Furthermore let $\mathcal{A}_{X_{T_1}} \subseteq X_{T_1}$ be an arbitrary assignment. We now construct an assignment $\mathcal{A}_{X_{T_2}}$, such that

$$\mathcal{A} = \mathcal{A}_{X_R} \cup \mathcal{A}_{X_{T_1}} \cup \mathcal{A}_{X_{T_2}} \models \Psi_{s,t}$$

holds. Without adding anything to $\mathcal{A}_{X_{T_2}}$, clearly $\mathcal{A} \models (s, t)$ and $\mathcal{A} \models (u, v) \Rightarrow \psi_{u,v}$ for all $(u, v) \in X_R \cap \bar{R}$ hold.

Let now $(u, v) \in R$ be an arbitrary pair of states. If $\mathcal{A} \not\models \pi_u(\Phi(u))$, then $\mathcal{A} \models \psi_{u,v}$ and $\mathcal{A} \models (u, v) \Rightarrow \psi_{u,v}$. Hence we assume now $\mathcal{A} \models \pi_u(\Phi(u))$. Since $(u, v) \in R$, there exists for all $M \in \text{Tran}_\emptyset(u)$ a set N , such that the condition holds, which is included in the assignment $\mathcal{A}_{X_{T_2}} \supseteq \pi_{u,v}(N)$. This can safely be done due to the prefixing and with Lemma A.1 we get $\mathcal{A} \models \varphi_{u,v}$ and $\mathcal{A} \models (u, v) \Rightarrow \psi_{u,v}$.

As a valuation \mathcal{A} can be constructed for a fixed modal refinement relation, such that for all subsets of X_{T_1} it satisfies the formula, $\exists X_R \forall X_{T_1} \exists X_{T_2} \Psi_{s,t} \in QBF_3^\exists$ holds.

‘Only-If’ part (correctness of the construction). We now assume

$$\exists X_R \forall X_{T_1} \exists X_{T_2} \Psi_{s,t} \in QBF_3^\exists$$

Then there exists a partial valuation $\mathcal{A}_{X_R} \subseteq \mathcal{A}$ for X_R , which satisfies $\Psi_{s,t}$. R is simply constructed by setting $R = \mathcal{A}_{X_R}$. Clearly $(s, t) \in R$. Let now $(u, v) \in R$ be an arbitrary pair of states. As (4) is satisfied for this pair, either $\Phi(u)$ is unsatisfiable and there simply exists no $M \in \text{Tran}_\emptyset(s)$ or for the chosen $M = \pi_u^{-1}(\mathcal{A}_{X_{T_1}})$ exists a $N = \pi_{u,v}^{-1}(\mathcal{A}_{X_{T_2}})$. By Lemma A.1 the modal refinement condition holds for this arbitrary pair. Hence R is a modal refinement relation.

Polynomial Runtime of the Reduction We show that the reduction indeed takes only polynomial time. For this observe that (5) is in $\mathcal{O}(|T_1(u)| \parallel |T_2(v)|)$. Therefore (4) is in $\mathcal{O}(|T_1(u)| \parallel |T_2(v)| + |\Phi_1(u)| + |\Phi_2(v)|)$. Leading to a total formula size of

$$\mathcal{O}(|S_1| \parallel |S_2| \parallel (|T_1| \parallel |T_2| + |\Phi_1| + |\Phi_2|))$$

B Thorough Refinement

B.1 Pruning

Now the preprocessing is formally introduced. Basically, we prune all the “inconsistent” states.

Definition B.1 (Consistency). *A state s of a BMTS is called locally consistent if $\Phi(s)$ is satisfiable, otherwise it is called locally inconsistent. If all states of a BMTS are locally consistent, the BMTS is called locally consistent. A state s of a BMTS is called globally consistent if it has an implementation, i.e. $\llbracket s \rrbracket \neq \emptyset$.*

Lemma B.2. *If (S, T, \emptyset, Φ) is a globally consistent BMTS, then for all $s \in S$:*

$$\forall M \in \text{Tran}_\emptyset(s) : \exists I \in \llbracket s \rrbracket : T_I(s) = M$$

Proof. Assume the conditions of the lemma. As the BMTS is globally consistent, for all $s \in S$ the set $\text{Tran}_\emptyset(s)$ is non-empty. Let now $s \in S$ be an arbitrary state and $M \in \text{Tran}_\emptyset(s)$ an arbitrary set of admissible transitions. We define

an implementation $(S_I, T_I, \emptyset, \Phi_I)$ with $S_I = \{t_I \mid t \in S\}$, $T_I(s_I) = M$ and for all $t_I \in S \setminus \{s_I\}$ and some $N \in \text{Tran}_\emptyset(t_I)$ we set $T_I(t_I) = N$. As e.g. $R = \{(t_I, t) \mid t \in S\}$ is a suitable modal refinement relation, $s_I \leq_m s$ holds.. \square

Corollary B.3. *If a state of a BMTS is locally consistent, it is also globally consistent.*

Proof. As the system is globally consistent, lemma B.2 is applicable. Because $\text{Tran}_\emptyset(s)$ is non-empty for every $s \in S$, there is at least one implementation refining s . Thus $\llbracket s \rrbracket \neq \emptyset$. \square

As one may have already noted, a locally inconsistent $s \in S$ of some system cannot have any implementation, as $\text{Tran}_\emptyset(s)$ is empty. This is captured by the following lemma.

Lemma B.4. *Removing a locally inconsistent state $s \in S$ from a BMTS does not change the semantic $\llbracket t \rrbracket$ of any other process $t \in S \setminus \{s\}$*

Proof. As the obligation of the state s is unsatisfiable, $\text{Tran}_\emptyset(s)$ is empty. Hence the modal refinement condition is always violated if the left system is locally consistent, which holds for implementations. Therefore, $(u, s) \notin R$ for any state u of an implementation. Removing the state from the system never affects the modal refinement relation, thus never changes the semantic of any other process of the system. \square

However, please note that while removing states from a system does not affect the semantic of other states, it still can make them locally inconsistent. As a preprocessing step, before constructing the *Avoid* relation, one has to remove all locally inconsistent states until the system becomes globally consistent. If one of the states, for which thorough refinement should be decided, is removed, the decision becomes trivial. If the the left one is inconsistent, the refinement holds. In the other case it does not.

B.2 Bounded Refinement

In the course of the proof of Lemma 5.9, we use a *bounded* version of definition 3.2, which coincides in the limit with the normal definition of modal refinement.

Definition B.5 (Bounded Modal Refinement). *Let $\mathcal{M}_1 = (S_1, T_1, P_1, \Phi_1)$ and $\mathcal{M}_2 = (S_2, T_2, P_2, \Phi_2)$ be two PMTS. A binary relation $R_{\mu, \nu}^n \subseteq S_1 \times S_2$ ($n \in \mathbb{N}_0$) is a bounded modal refinement relation under two fixed valuations $\mu \subseteq P_1$ and $\nu \subseteq P_2$ if either $n = 0$, then $R_{\mu, \nu}^0 = S_1 \times S_2$, or if for every $(s, t) \in R_{\mu, \nu}^{n+1}$ holds*

$$\begin{aligned} \forall M \in \text{Tran}_\mu(s) : \exists N \in \text{Tran}_\nu(t) : \\ \quad \forall (a, s') \in M : \exists (a, t') \in N : (s', t') \in R_{\mu, \nu}^n \\ \wedge \quad \forall (a, t') \in N : \exists (a, s') \in M : (s', t') \in R_{\mu, \nu}^n \end{aligned}$$

We say that s n -bounded modally refines t , denoted by $s \leq_m^n t$, if for all $\mu \subseteq P_1$ there exists a modal refinement relation $R_{\mu,\nu}^n$ with some $\nu \subseteq P_2$ such that $(s, t) \in R_{\mu,\nu}^n$.

Lemma B.6. *On finite PMTS modal refinement and bounded modal refinement coincide, meaning $s \leq_m t$ if and only if $s \leq_m^n t$ for all $n \in \mathbb{N}_0$.*

Proof. 'If' part. Let's assume $s \leq_m^n t$ for all $n \in \mathbb{N}_0$. Then there exists for every $\mu \subseteq P_1$ a nonincreasing series of sets $R_{\mu,\nu}^0, R_{\mu,\nu}^1, R_{\mu,\nu}^2 \dots$ with the *bounded* modal refinement definition applied each time and in all these sets is (s, t) contained. Every iteration of the *bounded* modal refinement definition will either remove at least one element or remove nothing and stabilize. As the underlying PMTS is finite, this series is stable after at most $|S_1 \times S_2|$ iterations and $R_{\mu,\nu} = R_{\mu,\nu}^{|S_1 \times S_2|} \ni (s, t)$ is a sufficient modal refinement relation for μ and ν . As this is applicable for every $\mu \subseteq P_1$, $s \leq_m t$ holds.

'Only-If' part. Let's assume $s \leq_m t$. Then there exists for every $\mu \subseteq P_1$ a modal refinement relation with $(s, t) \in R_{\mu,\nu}$. Let now $R_{\mu,\nu}^0, R_{\mu,\nu}^1, R_{\mu,\nu}^2 \dots$ be a nonincreasing series of sets with each time the *bounded* modal refinement definition applied. Clearly for all $i \in \mathbb{N}_0 : (s, t) \in R_{\mu,\nu} \subseteq R_{\mu,\nu}^i$. As this can be done for every $\mu \subseteq P_1$, we have $s \leq_m^n t$ for all $n \in \mathbb{N}_0$. \square

B.3 Proof of Lemma 5.9

First, we state a trivial technical claim.

Claim. *Avoid* is downward closed, i.e.

$$(s, \mathcal{T}) \in \text{Avoid} \implies \forall \mathcal{T}' \subseteq \mathcal{T} : (s, \mathcal{T}') \in \text{Avoid}$$

Proof (of Lemma 5.9). 'If' part (soundness of the construction). As *Avoid* is defined as smallest set, let $\text{Avoid}_0, \text{Avoid}_1, \text{Avoid}_2 \dots$ denote the non-decreasing sequence of sets leading to *Avoid* by applying the definition each time. We initialize Avoid_0 with (s, \emptyset) for all $s \in S$. We prove by induction on n that, whenever $(s, \mathcal{T}) \in \text{Avoid}_n$, there exists an implementation I such that $I \leq_m s$ and $\forall t \in \mathcal{T} : I \not\leq_m t$.

The base case $n = 0$ is trivial, as $\mathcal{T} = \emptyset$ the underlying BMTS is globally consistent and by corollary B.3 there is an implementation I for s . For the induction step assume $(s, \mathcal{T}) \in \text{Avoid}_{n+1}$.

As $(s, \mathcal{T}) \in \text{Avoid}_{n+1}$ there exists sets $M \in \text{Tran}_\emptyset(s)$ and $\text{later}_{a,s',f}$ such that the conditions of the definition hold. By the second part of the condition and the induction hypothesis, for all $(a, s') \in M$ and f there exists an implementation $I_{a,s',f}$ with $I_{a,s',f} \leq_m s'$ and $I_{a,s',f} \not\leq_m t'$ for all $t' \in \text{later}_{a,s',f}$. We now construct a new implementation I , such that $I \leq_m s$ and $I \not\leq_m t$ for all $t \in \mathcal{T}$. We simply take the disjoint union of the previously mentioned $I_{a,s',f}$, add a new state I with new transitions $(I, a, I_{a,s',f})$ for every $(a, s') \in M$ and f .

We now show that indeed $I \leq_m s$ and $I \not\leq_m t$ for all $t \in \mathcal{T}$ holds. The first claim trivially holds by construction. For the second claim, let us consider some

arbitrary $t \in \mathcal{T}$. Then for each $N \in \text{Tran}(t)$ there exists a particular action $a \in \Sigma$, for which one of the disjunctions holds.

Whenever the first is true then either $M(a)$ is empty, which is a violation of the modal refinement condition, or there exists $t' \in N(a)$, which is contained in $\text{later}_{a,s',f}$ for each $s' \in M(a)$. Since $I_{a,s',f} \not\leq_m t'$ the modal refinement condition is violated.

Whenever the second is true then again either $N(a)$ is empty, which is a direct violation of the modal refinement definition, as t cannot match the move of I , or there exists a $s' \in M(a)$, such that $\emptyset \neq N(a) \subseteq \text{later}_{a,s',f}$. Since $I_{a,s',f} \not\leq_m t'$ for all $t' \in \text{later}_{a,s',f}$, $(I_{a,s',f}, t')$ is never contained in a modal refinement relation. As $I_{a,s',f}$ is by construction an a -successor of I , the modal refinement condition is violated for (I, t) . Therefore the second claim holds.

'Only-If' part (completeness of the construction). We prove by induction on n , that whenever there exists an implementation I with $I \leq_m s$ and $I \not\leq_m^n t$ for all $t \in \mathcal{T}$ then $(s, \mathcal{T}) \in \text{Avoid}$. After that, lemma B.6 is applied. The base case $n = 0$ is trivial, as all pairs of processes are refining each other, hence $\mathcal{T} = \emptyset$ and by definition $(s, \mathcal{T}) \in \text{Avoid}$.

For the induction step assume the existence of an implementation I , such that $I \leq_m s$ and $I \not\leq_m^{n+1} t$ for all $t \in \mathcal{T}$. As I is an implementation, $\text{Tran}(I)$ is a singleton and $N_I \in \text{Tran}(I)$ is unique. Furthermore as $I \leq_m s$ holds, there exists by definition $N_s \in \text{Tran}(s)$, such that for all $a \in \Sigma$

1. $\forall s_a \in N_s(a) : \exists I_a \in N_I(a) : I_a \leq_m s_a$
2. $\forall I_a \in N_I(a) : \exists s_a \in N_s(a) : I_a \leq_m s_a$

To show that $(I, \mathcal{T}) \in \text{Avoid}_{n+1}$ we set $M := N_s$. For each $f \in \bigcup_{t \in \mathcal{T}} \text{Tran}(t)$ and each $(a, s') \in N_s$, we define $\text{later}_{a,s',f}$ such that the conditions are satisfied. We set

$$\begin{aligned} \text{later}_{a,s',f} := \{t' \mid \exists t \in \mathcal{T} : \exists N_t \in \text{Tran}(t) : t' \in N_t(a) \wedge \\ \forall I' \in N_I(a) : I' \not\leq_m^n t' \\ \vee f = N_t \wedge \exists I' \in N_I(a) : I' \leq_m s' \wedge \forall t'' \in N_t(a) : I' \not\leq_m^n t''\} \end{aligned} \quad (*)$$

Let $t \in \mathcal{T}$ and $N \in \text{Tran}(t)$ be arbitrary but fixed. As $I \not\leq_m^{n+1} t$, for some $a \in \Sigma$, there is a violation of the modal refinement definition, such that one of the cases hold:

1. $N_I(a) = \emptyset \wedge N_t(a) \neq \emptyset$
2. $N_I(a) \neq \emptyset \wedge N_t(a) = \emptyset$
3. $\exists t_a \in N_t(a) : \forall I_a \in N_I(a) : I_a \not\leq_m^n t_a$
4. $\exists I_a \in N_I(a) : \forall t_a \in N_t(a) : I_a \not\leq_m^n t_a$

If the third holds, then due to the first disjunct of $(*)$ we can satisfy the first disjunct of Definition by giving the same t_a . If the fourth holds, then due to the second disjunct of $(*)$ we can satisfy the second disjunct of Definition for any s' with $I' \leq_m s'$ (there is one due to 2.).

Finally, to prove that $(s_a, \text{later}_{a,s_a,f})$ has a n -step distinguishing implementation it is sufficient to take I' of the second disjunct.

□

C Thorough vs. Modal Refinement

C.1 Proof of Proposition 6.2

Proof. We fix a valuation ν of parameters and define a relation R that satisfies the condition of Definition 3.2. The relation R is taken as the smallest relation such that $(s_0', t_0) \in R$ and whenever $(s, t) \in R$, $(s, a, s') \in T$ and $(t, a, t') \in T$ then also $(s', t') \in R$. Before we prove that R satisfies the conditions, we make the claim that $(s, t) \in R$ implies $s \leq_t t$. Clearly, this holds for (s_0, t_0) . Suppose now that $s \leq_t t$, $(s, a, s'), (t, a, t') \in T$ and i' is an arbitrary implementation of s' . Then there exists an implementation $i \in \llbracket s \rrbracket$ such that $i \xrightarrow{a} i'$. But as $s \leq_t t$, i is also an implementation of t . Therefore, as t is deterministic, i' is an implementation of t' , thus $s' \leq_t t'$. We can now check that R satisfies the condition of Definition 3.2. Let $(s, t) \in R$ and $M \in \text{Trans}$. Define $A := \{a \mid \exists s' : (a, s') \in M\}$. There is an implementation i with exactly transitions under A . Moreover, according to the assumption it also an implementation of t . Hence $N := \{(a, t') \mid (t, a, t') \in T \wedge a \in A\}$ is an element of $\text{Tran}(t)$. The two conjuncts then clearly hold by construction of R . \square

C.2 Proof of Proposition 6.6

Proof. As the transition system of $\mathcal{D}(\mathcal{M})$ is created by the powerset construction, it is clearly deterministic. We prove that $s_0 \leq_m \mathcal{D}(s_0)$. Since both systems have the same parameter set, for any valuation of parameters of \mathcal{M} we can choose the same valuation for $\mathcal{D}(\mathcal{M})$. Further, we define relation R such that $(s, S) \in R$ iff $s \in S$ and show that the condition of Definition 3.2 is satisfied. Let $(s, S) \in R$. For $M \in \text{Tran}(s)$, we set $N := M[(a, S_a)/(a, s')]$. Since $\text{Tran}(S) = \bigcup_{s \in S} \text{Tran}(s)[(a, S_a)/(a, s')]$, we have $N \in \text{Tran}(S)$. We check the two conjuncts. Whenever there is $(a, s') \in M$ then $(a, S_a) \in N$ and $s' \in S_a$ hence $(s', S_a) \in R$. Whenever there is $(a, S_a) \in N$ we have the respective $(a, s') \in M$ with by construction of N . Further, $s' \in S_a$ hence $(s', S_a) \in R$. \square

C.3 Proof of Proposition 6.7

Proof. Assume t_0 deterministic state of a PMTS \mathcal{N} with $s_0 \leq_m t_0$. Therefore, there for every valuation μ there is a valuation ν and the greatest relation $R_{\mu, \nu}$ containing (s_0^μ, t_0^ν) and satisfying the condition of Definition 3.2. We show that $\mathcal{D}(s_0) \leq_m t_0$ by choosing for every μ the same ν and constructing a new relation $Q_{\mu, \nu}$ between states of $\mathcal{D}(\mathcal{M})^\mu$ and \mathcal{N}^ν that also satisfies this condition as follows:

$$(S, t) \in Q_{\mu, \nu} \quad \text{if and only if} \quad \emptyset \neq S \subseteq \{s \mid (s, t) \in R_{\mu, \nu}\}$$

We now check the condition. Since $(s_0, t_0) \in R_{\mu, \nu}$, we have $(\mathcal{D}(s_0)^\mu, t_0^\nu) = (\{s_0\}^\mu, t_0) = (\{s_0^\mu\}, t_0) \in Q_{\mu, \nu}$. Let now $(S, t) \in Q_{\mu, \nu}$ and $M \in \text{Tran}(S)$. Hence there is $s \in S$ with $M' \in \text{Tran}(s)$ with $M = M'[(a, S_a)/(a, s')]$ for every a, s' .

Since $(s, t) \in R_{\mu, \nu}$, there is $N \in \text{Tran}(t)$ matching M' . We show it also matches M . Let $(a, S_a) \in M$. There is unique (due to determinism) $(a, t') \in N$ and further $(S_a, t') \in Q_{\mu, \nu}$ as each $s_a \in S_a$ modally refines the only a -successor of t , thus $(s_a, t') \in R_{\mu, \nu}$. Similarly, let $(a, t') \in \text{Tran}(t)$. Then there is unique $(a, S') \in \text{Tran}(S)$, namely (a, S_a) . For the same reasons as above $(s_a, t') \in R_{\mu, \nu}$ for every $s_a \in S_a$.

The minimality for BMTS holds w.r.t. both thorough and modal refinements as they coincide when the refined system is a deterministic BMTS. \square

C.4 Proof of Proposition 6.9

Proof. First, observe that for any parameter valuation ν , the identity relation satisfies the condition of Definition 3.2 for s_0^ν and $\mathcal{P}(s_0)$. Indeed, for any $M \in \text{Tran}(s)$ we also have $M \in \text{Tran}(\mathcal{P}(s))$. Similarly, $\{((s, \nu), s) \mid s \in S, \nu \subseteq P\} \cup \{(s_0^B, \mathcal{P}(s_0))\}$ satisfies the condition for s_0^B and $\mathcal{P}(s_0)$. \square